UNITED STATES PATENT APPLICATION

FOR

A PLATFORM AND METHOD OF CREATING A SECURE BOOT
THAT ENFORCES PROPER USER AUTHENTICATION
AND ENFORCES HARDWARE CONFIGURATIONS

INVENTOR:
David W. Grawrock

PREPARED BY:
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(714) 557-3800

A PLATFORM AND METHOD OF CREATING A SECURE BOOT THAT ENFORCES PROPER USER AUTHENTICATION AND ENFORCES HARDWARE CONFIGURATIONS

5 ## BACKGROUND

1. ### FIELD

This invention relates to the field of data security. In particular, the invention relates to a platform and method for protecting information through a secure boot using user authentication and/or hardware configurations.

10 2. ### BACKGROUND

Personal computers (PCs) typically include different types of storage components to store programs and data. These storage components include random access memory (RAM), read-only memory (ROM), and memory devices that are located external to the PC (e.g., hard disk or a floppy disk). To load an operating

15 system on a PC, it is necessary to initialize or "boot" the PC by loading and executing boot code. Because the PC typically is unable to access external devices until after it is booted, the boot code is stored internally within the PC.

Typically, a ROM component is used to store the boot code. This boot code, normally referred to as "boot block," is obtained from the ROM and executed. The

20 boot block is coded to (i) locate Basic Input/Output System (BIOS), (ii) load the BIOS for execution, and (iii) pass control to the BIOS. Thereafter, the BIOS checks Option ROMs, loads the operating system (OS) loader, and passes control to the OS loader.

Currently, enhanced security features are being implemented in platforms with greater regularity. However, current security features lack the combination of both user

25 authentication and secure boot functionality.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is an exemplary embodiment of a platform utilizing the present

5   invention.

Figure 2 is an exemplary embodiment of the IC device configured as a Trusted Platform Module (TPM) employed within the platform of Figure 1.

Figure 3 is an exemplary embodiment of the TPM of Figure 2.

Figures 4A-4B are exemplary embodiments of binding operations performed by

10   the TPM of Figure 3.

Figure 5 is an exemplary embodiment of a flowchart illustrating the operations during initialization of the platform of Figure 1.

Figure 6 is an exemplary embodiment of a block diagram illustrating the operations during initialization of the platform as shown in Figure 5.

DESCRIPTION

The present invention relates to a platform and method for protecting information through a secure boot process using user authentication and/or hardware configurations. More specifically, the invention comprises the employment of one or

5   more additional boot operations into a boot process in order to enhance security; namely, (i) the binding of a segment of Basic Input/Output System (BIOS) code to its platform and current configuration (e.g., hardware configuration within the platform) and (ii) the encryption of another segment of the BIOS code using binding operations and contents of a token to recover keying material.

10   Herein, certain details are set forth in order to provide a thorough understanding of the present invention. It is apparent to a person of ordinary skill in the art, however, that the present invention may be practiced through many embodiments other that those illustrated. Well-known circuits are not set forth in detail in order to avoid unnecessarily obscuring the present invention.

15   In the following description, certain terminology is used to discuss features of the present invention. For example, a "platform" includes any product that performs operations for subsequent analysis and verification of the platform's boot process. Examples of a platform include, but are not limited or restricted to a computer (e.g., desktop, a laptop, a server, a workstation, a personal digital assistant, etc.) or any

20   peripherals associated therewith; communication equipment (e.g., telephone handset, pager, etc.); a television set-top box and the like. A "link" is broadly defined as a logical or physical communication path such as, for instance, electrical wire, optical fiber, cable, bus trace, or even a wireless channel using infrared, radio frequency (RF), or any other wireless signaling mechanism.

25   In addition, the term "information" is defined as one or more bits of data, address, and/or control. "Code" includes software or firmware that, when executed, performs certain functions. Examples of code include an application, an applet, or any other series of instructions. "Keying material" includes a cryptographic key or information used to produce a cryptographic key.

30   A "cryptographic operation" is an operation performed to enhance data security through obfuscation, integrity protection and the like. For example, one type of cryptographic operation is hashing, namely a one-way conversion of information to a fixed-length representation that is referred to as a hash value. Normally, the "hash value" is substantially lesser in size than the original information. It is contemplated

-3-

that, in some cases, the hashing may involve a 1:1 conversion. One type of hashing function is referred to as The Secure Hash Algorithm (SHA-1) as specified by The National Institute of Standards of Technology.

5   A "binding" operation is the act of performing operations, within a device, to obfuscate information and subsequently recover the information using a secret value stored inside the device and/or a token of that device. In one embodiment, the binding operation involves a combination of encryption and non-volatile storage that creates encrypted information that can only be decrypted using the secret value. Herein, a "token" is any type of device that can securely store information. As an optional

10   characteristic, the token may be removable from the platform. Illustrative examples of the token include, but are not limited or restricted to a smart card, a PCMCIA card, a Bluetooth(tm) device, a Universal Serial Bus (USB) token, and the like.

When the binding operation is performed during a boot process, the secret value is never revealed outside the device. In one situation, the secret value may be a

15   cryptographic key. For another situation, however, the secret value may be computed integrity metrics for the platform. An "integrity metric" is a cryptographic hash value of information such as the BIOS, option ROM (e.g., BOIS extension) or another type of information. One configuration of an integrity metric, referred to as the "hardware metric," can be stored within one or more internal registers and represent the

20   configuration of one or more hardware devices of the platform.

Referring to Figure 1, an exemplary block diagram of an illustrative embodiment of a platform 100 employing the present invention is shown. The platform 100 comprises a processor 110, a memory control hub (MCH) 120, a system memory 130, an input/output control hub (ICH) 140, and an integrated circuit (IC)

25   device 150 which initiates, monitors and controls the boot process of the platform 100.

As shown in Figure 1, the processor 110 represents a central processing unit of any type of architecture, such as complex instruction set computers (CISC), reduced instruction set computers (RISC), very long instruction word (VLIW), or a hybrid architecture. In one embodiment, the processor 110 is compatible with the INTEL(r)

30   Architecture (IA) processor, such as the IA-32 and the IA-64. Of course, in an alternative embodiment, the processor 110 may include multiple processing units coupled together over a common host bus 105.

Coupled to the processor 110 via the host bus 105, the MCH 120 may be integrated into a chipset that provides control and configuration of memory and

input/output (I/O) devices such as the system memory 130 and the ICH 140. The system memory 130 stores system code and data. The system memory 130 is typically implemented with dynamic random access memory (DRAM) or static random access memory (SRAM).

5　　　　The ICH 140 may also be integrated into a chipset together or separate from the MCH 120 to perform I/O functions. As shown, the ICH 140 supports communications with the IC device 150 via link 160. Also, the ICH 140 supports communications with components coupled to other links such as a Peripheral Component Interconnect (PCI) bus at any selected frequency (e.g., 66 megahertz "MHz", 100 MHz, etc.), an Industry

10　Standard Architecture (ISA) bus, a Universal Serial Bus (USB), a Firmware Hub bus, or any other bus configured with a different architecture than those briefly mentioned. For example, the ICH 140 may be coupled to non-volatile memory 170 (e.g., flash memory) to contain BIOS code.

The non-volatile memory 170 includes BIOS code, portions of which have

15　undergone a binding operation. For instance, a first BIOS area (BIOS Area 1) 171 is a first segment of the BIOS code that has undergone a binding operation. This binding operation ensures that the first BIOS segment cannot be executed on any other platform besides platform 100 and that the platform 100 employs certain hardware that was in place during the binding operation. A second BIOS area (BIOS Area 2) 172 is another

20　(second) segment of the BIOS code that has undergone a binding operation. This binding operation ensures that (i) a selected user has authorized use of platform 100, and (ii) user authorization has occurred on the same platform and within the same hardware and BIOS configuration as described in Figure 5.

Of course, it is contemplated that the IC device 150 may be employed in a

25　different embodiment than described above. For example, although not shown, the functionality of the IC device 150 may be employed within the ICH 140. Thus, any packaging associated with the ICH 140 would protect the IC device from damage caused by contaminants or unauthorized probing.

Referring to Figure 2, an exemplary embodiment of the IC device 150 is shown.

30　The IC device 150 comprises one or more integrated circuits placed within a protective package 200 such as any type of integrated circuit package, a cartridge covering a removable daughter card featuring the integrated circuit(s) and the like. For this embodiment, the IC device 150 comprises a boot block memory unit 210 in communication with logic 220 that performs various binding and cryptographic

-5-

operations. For instance, the logic 220 may be implemented as a trusted platform module (TPM) as described in Figure 3.

Referring now to Figure 3, an exemplary embodiment of a TPM is shown. The TPM comprises at least an I/O interface 300, a processor 310, and internal memory 320
5　(e.g., volatile and/or non-volatile). Herein, the processor 310 is configured to access certain content within the internal memory 320 (e.g., software, keying material, etc.) to perform cryptographic operations on incoming information. One of these cryptographic operations includes a binding operation as shown in Figures 4A-4B. Of course, in lieu of the processor 310 performing the cryptographic operations, it is
10　contemplated that a cryptographic unit separate from the processor 310 may be employed.

For instance, as shown in Figure 4A, the first BIOS segment 400 undergoes a binding operation by encrypting this selected segment of the BIOS code using keying material 410 securely stored in the internal memory of the TPM. This binding
15　operation is performed in a controlled environment such as by an original equipment manufacturer or by an entity authorized by a manufacturer of the TPM or platform (e.g., an authorized retailer or distributor, an information technology "IT" department of a business, etc.)

In addition, this binding operation could further utilize an integrity metric (e.g.,
20　a hardware metric) as an additional binding parameter 440 as shown by dashed lines. The hardware metric may be a hash value of identification (ID) information for certain hardware employed within the platform. The "ID information" may be a pre-stored serial number, a hash value of a serial number or some other type of static information such as driver code from an IDE controller, a Network Interface Card (NIC) address
25　and the like. This provides an additional check that the platform 100 has the same hardware as when the first BIOS segment 400 was "bound" to the platform.

In addition, as shown in Figure 4B, another (second) segment of the BIOS code 450 undergoes a binding operation by encrypting that segment using a key 460 formed by a combination of both (1) keying material 470 provided by a token associated with
30　the platform and (2) keying material 480 stored within internal memory of the TPM (referred to as the "combination key" or "C_Key").

Referring to Figures 5 and 6, a flowchart and corresponding block diagram illustrate the operations during initialization of the platform of Figure 1. It is contemplated that the binding of the first BIOS segment is performed to ensure that the

platform and its hardware configuration have not been modified. The binding of the second BIOS segment is to ensure that the user has authorized use of the platform and that the platform is still implemented with the same hardware and BIOS configurations.

Initially, as shown in item 500, the boot block loads the BIOS code into the
5 TPM. The BIOS code includes at least BIOS Area 1 and BIOS Area 2. The amount of BIOS code associated with the first BIOS segment (BIOS Area 1) needs to be sufficient to enable communications with the TPM and that the TPM is able to perform an unbinding operation. For instance, the first BIOS segment may include a section of the BIOS code that executes immediately after the BIOS gets address ability to the
10 TPM.

Thereafter, as shown in item 510, the TPM recovers keying material for use in recovering the first BIOS segment from BIOS Area 1. Herein, the key is used to decrypt BIOS Area 1 either within the TPM or in the BIOS itself (item 520). "Decryption" is represented through a "DEC( )" label. It is contemplated that if a
15 hardware metric is used as an additional binding parameter, the binding would occur after creation of the hardware metric and would require an additional check that the platform has the same hardware as when the first BIOS segment was bound to the platform. This check may be accomplished, for example, by (i) performing a hashing operation on ID information from selected hardware in order to produce a result and (ii)
20 comparing the result with the hardware metric. Normally, the hardware metric is stored in platform configuration registers within the platform.

Thereafter, the BIOS process continues until a user authentication sub-process is encountered (item 530). Upon encountering the user authentication sub-process, the BIOS determines at least whether the user has been authenticated through an acceptable
25 authentication mechanism as shown in item 540. Acceptable authentication mechanisms include, for example, a password-based mechanism or a biometrics mechanism (e.g., fingerprint scan, retinal scan, hand or face geometry scan, and the like).

If authentication (item 550), the token releases information to the TPM. In this
30 embodiment, the information is keying material (referred to as a "first-half key"). Otherwise, the first-half key is not released by the token, which prevents subsequent recovery of the second BIOS segment through decryption of BIOS Area 2 (item 560).

The first-half key is provided to the TPM, which produces a combination key as shown in items 570 and 580. In this embodiment, the combination key is produced by

-7-

performing a key combination operation on both the first-half key and keying material stored within the internal memory of the TPM (referred to as "second-half key"). Examples of a "key combination operation" include hashing, a bitwise exclusive OR (XOR), encryption, addition, subtraction, concatenation and the like. Thereafter, the second BIOS segment is recovered using the combination key (e.g., via decryption) as shown in item 590.

Concurrently or subsequent to recovering the second BIOS segment, as an optional operation, the TPM may unbind keying material associated with a non-volatile storage device (e.g., hard disk drive) as shown in item 595. In this embodiment, the keying material is "unbound" by decrypting it with the combination key and perhaps hardware metrics. This enables the user to access content stored on the non-volatile storage device.

It is contemplated that an optional enhancement of the present invention may involve on-the-fly key modification. This may be accomplished by implementing an access control mechanism to prevent unauthorized access to the second-half key stored in platform memory in lieu of storing the second-half key within internal memory of the TPM. The type of access control mechanism that may be implemented includes Isolated Execution (ISOX(tm)) techniques by Intel Corporation of Santa Clara, California. This would allow remote reset of the BIOS to a new value without having physical access to the platform.

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art. Additionally, it is possible to implement the present invention or some of its features in hardware, firmware, software or a combination thereof where the software is provided in a processor readable storage medium such as a magnetic, optical, or semiconductor storage medium.